

Ralph Stelzer · Karl-Heinrich Grote · Klaus Brökel
Frank Rieg · Jörg Feldhusen (Hrsg.)

ENTWERFEN ENTWICKELN ERLEBEN

Methoden und Werkzeuge in der Produktentwicklung



**10. Gemeinsames Kolloquium Konstruktionstechnik
KT2012 | Residenzschloss Dresden | 14.–15. Juni 2012**

Stelzer · Grote · Brökel · Rieg · Feldhusen (Hrsg.)

ENTWERFEN ENTWICKELN ERLEBEN

Methoden und Werkzeuge in der Produktentwicklung

10. Gemeinsames Kolloquium Konstruktionstechnik KT2012

Ralph Stelzer · Karl-Heinrich Grote · Klaus Brökel
Frank Rieg · Jörg Feldhusen (Hrsg.)

ENTWERFEN ENTWICKELN ERLEBEN

Methoden und Werkzeuge in der Produktentwicklung

Entwickeln – Entwerfen – Erleben.
Methoden und Werkzeuge in der Produktentwicklung
10. Gemeinsames Kolloquium Konstruktionstechnik (KT2012)

Herausgeber:

Prof. Dr. Ralph Stelzer (Technische Universität Dresden)
Prof. Dr. Karl-Heinrich Grote (Otto-von-Guericke-Universität Magdeburg)
Prof. Dr. Klaus Brökel (Universität Rostock)
Prof. Dr. Frank Rieg (Universität Bayreuth)
Prof. Dr. Jörg Feldhusen (RWTH Aachen)

Wir bedanken uns für die Unterstützung bei
ma design, Tedata, Continental, xPLM, B.I.M. Consulting und Reiss Büromöbel

ma design
//ENGINEERING

Continental 

B.I.M.
consulting

TEDATA

xPLM
Solution

REISS

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der
Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind
im Internet über <http://dnb.d-nb.de> abrufbar.

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche
Nationalbibliografie; detailed bibliographic data are available in the
Internet at <http://dnb.d-nb.de>.

ISBN 987-3-942710-80-0

© 2012 TUDpress
Verlag der Wissenschaften GmbH
Bergstr. 70 | D-01069 Dresden
Tel.: 0351/47 96 97 20 | Fax: 0351/47 96 08 19
<http://www.tudpress.de>

Alle Rechte vorbehalten. All rights reserved.
Layout und Satz: Sandra Olbrich/Technische Universität Dresden.
Umschlaggestaltung: TU Dresden, Illustration Audi A6 Limousine © 2012 Audi AG

Interdisziplinäre Produktentwicklung

1 Einleitung

Innovative interdisziplinäre Produktentwicklung erfordert ein Überdenken von heutigen Methoden, Prozessen, IT-Lösungen und Organisationsformen. Insbesondere fehlt es an Unterstützung durch geeignete IT-Lösungen für die funktionale Beschreibung und Auslegung von Systemarchitekturen. Für die disziplinübergreifende Systemmodellierung der Konzeptphase gibt es nur eingeschränkte IT-Unterstützung. Elektronik und Software stellen einen immer stärkeren Anteil im Produktentwicklungsprozess (PEP) dar. Konstruktions- und Entwurfsmethoden dieser Disziplinen sollten auf den Prüfstand gestellt und ihre Tauglichkeit für einen moderne interdisziplinären Konstruktionsansatz überprüft werden. Model Based Systems Engineering (MBSE) könnte sich als integrative Methode etablieren und eine Brücke zwischen den verschiedenen Ingenieurdisziplinen bilden. Als »Enabler« für das MBSE werden Systemmodellierungssprachen wie z.B. SysML vorgestellt, die ein Werkzeug für eine interdisziplinäre Systembeschreibung darstellen. Auf konkreterer Stufe können Simulationssprachen, wie z.B. Matlab/Simulink oder Modelica eine frühe multidisziplinäre Simulation ermöglichen, die in Verbindung mit Systembeschreibungssprachen eine frühe Konzeptformulierung erlauben. Dieser Beitrag soll neue Methoden, Prozesse und IT-Lösungen für eine interdisziplinäre virtuelle Produktentwicklung aufzeigen (Anderl et al. 2012).

Model Based Systems Engineering (MBSE) ist ein multidisziplinärer Ansatz mit dem Ziel, eine ausgewogene Systemlösung als Reaktion auf diverse Stakeholder Bedürfnisse zu entwickeln (Friedenthal 2009). MBSE hilft dem Ingenieur den Überblick über komplexe Systeme zu behalten und den Zusammenhang zu verstehen und die Spezifikation und damit alle definierten Anforderungen zu erfüllen. Während klassische Methoden des Systems Engineering papier- oder dokumentenbasiert sind, basiert MBSE auf digitalen Systemmodellen. Diese erlauben die Erfassung der Komplexität und erleichtern den Informationsaustausch unter Disziplinen. Modellierungssprachen ermöglichen eine modellbasierte Spezifikation eines Produktes auf einer allgemeinen Ebene die verständlich für alle an der Entwicklung beteiligten Disziplinen ist. Das Problem der Integration der Komponenten während des Entwicklungsprozesses kann durch die Verwendung solcher Modellierungssprachen möglichst früh in Angriff genommen werden, indem die Definition von Korrelationen zwischen Systemanforderungen, Funktionen, Struktur und Verhalten definiert wird.

Darüber hinaus werden in heutigen Unternehmen die im Entwicklungsprozess anfallenden Daten meist durch Product Lifecycle Management (PLM)-Lösungen verwaltet, so dass die Integration einer funktionalen Produktbeschreibung in PLM die interdisziplinäre Zusammenarbeit frühzeitig unterstützen kann.

2 Ausgangssituation

Die Realisierung der vielfältigen Funktionalitäten von Produkten hat sich in den letzten Jahren von rein mechanischen zu mechatronischen Komponenten verändert. Funktionen und Verhalten als eine neutrale Beschreibung werden mehr und mehr wichtig, wenn es um eine interdisziplinäre Entwicklung geht. Die Komplexität für die Integration mechatronischer Komponenten während des PEP steigt durch die starke Beteiligung der verschiedenen Disziplinen. Darüber hinaus spielt Software eine immer größere Rolle im Kontext der modernen interdisziplinären Produktentwicklung. Kommunizieren Produkte miteinander, wird von Cyber-Physical Systems bzw. cybertronischen Systemen gesprochen. Aktuelle Forschungsinitiativen konzentrieren

sich auf technologische Fortschritte mit Softwareintensiven eingebetteten Systemen in technischen Produkten (Broy 2010). Software wird in Zukunft eine Vielzahl von weiteren Produktfunktionen ermöglichen. Dies setzt eine noch stärkere Einbeziehung des Software-Engineering in den PEP voraus. Deshalb ist es wichtig, die Transparenz zwischen den Disziplinen zu erhöhen. Es fehlt jedoch an etablierten, d.h. industriell eingesetzten, Methoden, Prozessen und IT-Lösungen für die disziplinübergreifende Entwicklung interdisziplinärer Systeme und damit intelligenter und vernetzter Produkte und Produktionssysteme. Dabei hat sich gerade in Deutschland hat sich in den 60er und 70er Jahren eine auf Funktionen basierende Entwicklungsmethodik mit abgeleiteten Entwicklungsprozessen gebildet, die natürlich zu dieser Zeit noch nicht auf formalen Sprachen aufsetzte und schwerpunktmäßig mechanisch geprägt war. Parallel haben sich international im Wesentlichen auf formalen Sprachen basierende Software-Entwicklungsmethoden ausgebildet (siehe Abbildung 1).

Seit den 50er Jahren werden insbesondere bei der amerikanischen Luft- und Raumfahrt und in großen Militärprojekten Systems Engineering (SE) als interdisziplinärer dokumentengetriebener Ansatz, zur Entwicklung und Umsetzung komplexer technischer Systeme in großen Projekten definiert und eingesetzt. Dieser Ansatz wurde aus Sicht der Software- und Elektronikindustrie permanent ausgebaut und bietet heute Modellierungs- und Simulationsunterstützung von komplexen, stark vernetzten Systemen an.

Heute genutzte PLM und ERP Konzepte sind für die Umsetzung der Anforderungen aus der interdisziplinären Produkt- und Produktionssystementwicklung in der frühen Phase des PEP nicht geeignet. Die bisher genutzten Datenmodelle sind zu starr ausgelegt und die Trennung in monolithische Systeme für Entwicklung (PLM), Logistik, Produktion und Personal (PPS, ERP), Kunden- und Zuliefererintegration (CRM, SCM) sind nicht mehr zeitgemäß. Sie ermöglichen nicht mehr die Erfassung der komplexen Zusammenhänge der Produkte selbst, ihrer Wechselwirkung untereinander, sowie ihr Zusammenspiel insbesondere in der frühen Phase des Entwicklungsprozesses mit zunehmend intelligent auszulegenden Produktions- und Infrastruktur Systemen umzusetzen.

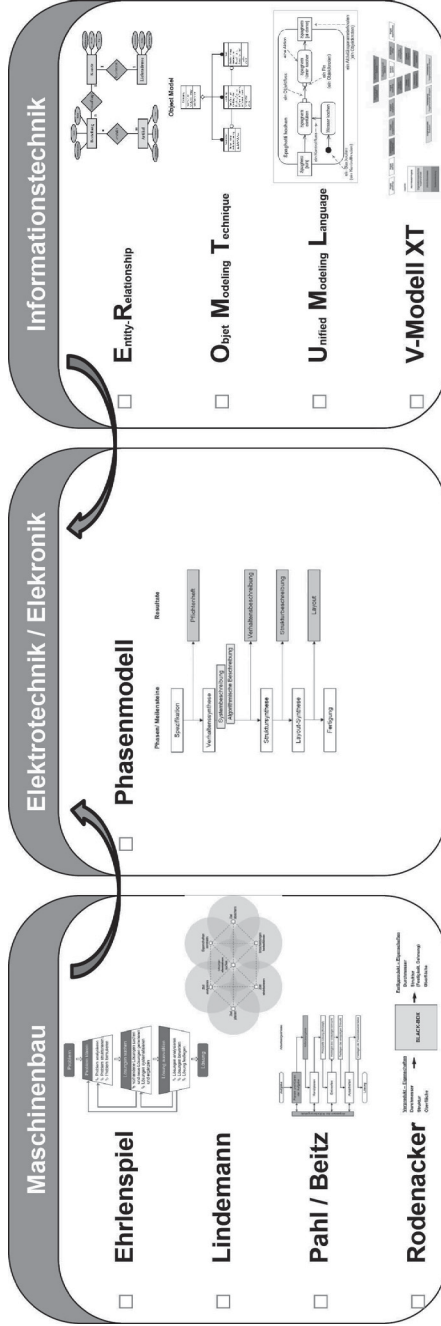


Abb. 1: Disziplinspezifische Entwicklungsmethoden und -prozesse

In den verschiedenen Phasen des PEP existieren bereits heute schon eine Vielzahl von Anwendungssprachen und -systemen:

- Modellbildung und Spezifikation: SysML (OMG Standard), ModellicaML
- Modellbildung und Simulation: Simulink, Simscape, Matlab, Modellica
- Disziplinspezifische Modellbildung und Simulation: hier existieren eine Vielzahl von M-CAD, E-CAD, CASE (Computer Aided Software Engineering) und CAE-Lösungen

Die IT-Lösungen der zweiten und insbesondere der dritten Phase sind weitgehend industriell implementiert und akzeptiert. Auf der ersten Ebene sind trotz Existenz internationaler Standards noch keine industriellen Anwendungen für interdisziplinäre Produkt- und Produktionssystementwicklung bekannt. Von wissenschaftlicher Seite existieren hier Vorschläge neben dem OMG Standard eigene Modellierungssprachen zu entwickeln. Die Schnittstellen zwischen den drei Ebenen sind nur teilweise existent und müssen für eine durchgängige Prozessgestaltung entwickelt, implementiert und ausgebaut werden. Zusammenfassend ist die Ausgangslage gekennzeichnet durch fehlende integrative Methoden, Prozesse und IT-Lösungen – sowohl auf der Administrationsseite (PLM und ERP) als auch auf der Anwendungsseite – bzw. deren fehlende Integration in ein durchgängiges Modellierungskonzept von Anforderungen (A), Funktionen (F), logischer (L) und physischer Beschreibung von interdisziplinären Produkten und Produktionssystemen. Der Ansatz des Systems Engineering und insbesondere des Model Based Design könnte ein zukünftiger Leitfaden für Methoden, Prozesse und IT-Lösungen zur Entwicklung interdisziplinärer Produkte und Produktionssystemen sein.

3 Stand der Forschung

In diesem Abschnitt werden die damit verbunden Arbeiten auf dem Gebiet der konzeptionellen Produktmodellierung andiskutiert. Dies beinhaltet die funktionalen Modellierung und Beschreibungssprachen von mechatronischen Systemen sowie darüber weitergehend auch die PLM Integration dieser Beschreibung.

3.1 Funktionale Modellierung

Die funktionale Modellierung stellt eine abstrakte Methode für das Verständnis und für den Gesamtwert eines Produktes dar. Die primäre Aufgabe ist die Unterstützung der Suche nach geeigneten Lösungen im Design, sowie die Schaffung von einem disziplinunabhängigen Modell eines Produktes. Ein funktionelles Modell enthält eine abstrakte Beschreibung der wichtigsten Ziele eines Produktes durch Angabe seiner Funktionen (Substantiv-Verb-Kombination) (Pahl et al. 2007). Die gesamte Funktion wird in seine Teilfunktionen unterteilt, von denen jede wiederum in weitere Teilfunktionen unterteilt werden können. Dies führt zu einer funktionalen Hierarchie. Eine typische Darstellung eines funktionalen Modells ist ein Diagramm mit den Funktionen als separate Blöcke und Verbindungen, wie z.B. Stoff-, Energie- oder Signalaustausch, dargestellt werden können.

Bisher wurden funktionale Modelle nicht im Sinne eines modellbasierten Ansatzes betrachtet, sondern mehr als Dokumente verstanden. Darüber hinaus sind funktionale Modelle selten formal und damit durch den Computer interpretierbar.

Ein Schritt in Richtung des modellbasierten Entwurfes ist der Ansatz von Stone (Stone et al. 2002). Es geht um die Definition eines formalisierten Verfahrens auf Basis von Taxonomien und Regeln für die Beschreibung von Funktionen, die das Produktdesign beschreiben.

3.2 Modellierungssprachen für die konzeptionelle Produktmodellierung

Gausemeier et.al. schlagen eine Reihe von Spezifikationstechniken für die Beschreibung von selbstoptimierenden mechatronischen Systemen vor. Mit Hilfe verschiedener Ansichten kann das System beschrieben werden. Jede Ansicht wird durch ein partielles Modell z.B. in Form eines Diagrammes dargestellt. In einem Teilmodell können spezifische Zusammenhänge dargestellt werden. Die wichtigsten Teilmodelle beschreiben Umfeld, Zielsystem, Verhalten, Anwendungsszenarien, Anforderungen, Funktionen, Wirkstruktur und Gestalt. Das Konzeptmodell wird durch die Summe von kohärenten Teilmodellen gebildet. (Gausemeier et al. 2009)

Eine andere von der OMG vorgeschlagene Sprache für die Beschreibung von Systemen ist SysML. SysML wird als Werkzeug zur unabhängigen grafischen Modellierung im Rahmen von Spezifikation, Analyse, Design, Verifikation und Validierung von Systemen verstanden. Aufbauend auf SysML und UML gab es Bestrebungen Simulationsmodelle zu integrieren. ModelicaML ist eine grafische Modellierungssprache basierend auf einem UML-Profil, um Simulationsmodelle in Modelica mit einer SysML ähnlicher Semantik zu beschreiben. ModelicaML ermöglicht die Erstellung von Beschreibungsmodellen nach den Methoden des modellbasierten Systems Engineering, die zudem in Modelica ausführbar sind (Schamai et al. 2009). Paredis et.al. schlagen darüber hinaus eine Spezifikation für die Modelltransformation von SysML-Modellen in Modelica Modellen vor, um so die analytischen Vorteile von Modelica mit der Beschreibungsfreiheit von SysML zu kombinieren (Paredis et al. 2010).

3.3 PLM Integrationsaktivitäten

Um den interdisziplinären Produktentwicklungsprozess durchgängig unterstützen zu können ist eine Integration in PLM sinnvoll. Zur Integration in eine PLM Lösung werden hauptsächlich gängige Standards betrachtet.

Die XML Metadata Interchange Spezifikation (XMI) standardisiert durch die OMG unterstützt den Austausch von Modelldaten zwischen den Werkzeugen für Meta Object Facility (MOF)-basierte Modellierungssprachen, definieren jedoch kein Datenschema. Ein weiterer Austauschstandard für den Austausch von Systems Engineering ist STEP AP233. In der SEDRES Projektgruppe wurde STEP AP233 für die Integration von Systems Engineering-Daten in PDM-Systemen formuliert. AP233 unterstützt die Systemmodellierung, welche die Systemstruktur und das Verhalten integriert (Eckert et al. 2005).

In der OMG untersucht weiterhin eine Arbeitsgruppe die Zusammenhänge zwischen SysML und AP233. Ziel ist ein Mapping der gemeinsamen Datenkonstrukte, sowie ein Change Management für SysML.

4 Erweiterter modellbasierter Ansatz in der Virtuellen Produktentwicklung

Die modellbasierte Entwicklung ist in der virtuellen Produktentwicklung von zentraler Bedeutung. Modelle können zum Beispiel topologische, physische, prozessorientierte, geometrische oder mathematische Modelle sein.

Im modellbasierten Systems Engineering (MBSE) werden Modelle für die Beschreibung und Spezifikation verwendet, um eine Strukturierung von komplexen technischen Problemen zu erleichtern. Dabei werden die Beziehungen zwischen Eigenschaften für die Analyse auf einer höheren strukturellen Ebene erfasst. Akteure aus verschiedenen Disziplinen sind in der Konzeption und Entwicklung eines komplexen Systems beteiligt. Jeder Stakeholder hat einen anderen Blick auf die Spezifikation (Friedenthal 2009). Die Metho-

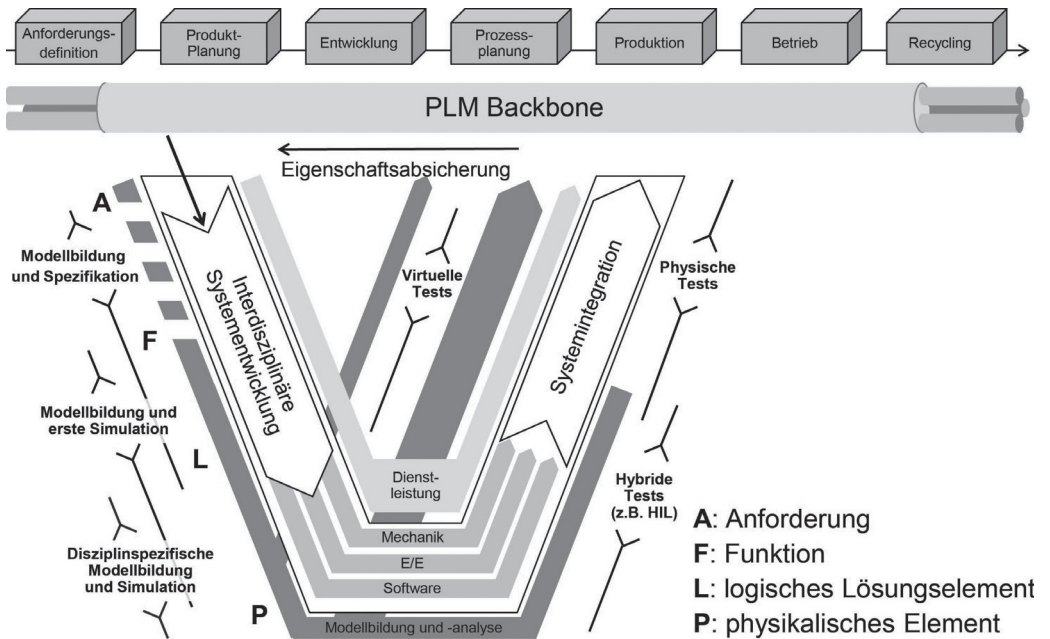


Abbildung 2: Erweitertes V-Modell für Model Based Systems Engineering

den des modellbasierten Systems Engineering können dazu beitragen, ein multidisziplinäres Produkt in einer abstrakten Weise zu beschreiben. Die VDI 2206 definiert einen systematischen Ansatz für die Entwicklung mechatronischer Systeme. Der Fokus in diesem Artikel liegt hier auf dem linken Flügel des »V« und erweitert es mit dem Einsatz von Methoden aus der modellbasierten Systems Engineering (siehe Abbildung 2).

Es können drei Ansichten der Modellierung identifiziert werden:

- Modellbildung und Spezifikation: Ein System wird durch qualitative Modelle beschrieben. Diese beinhalten Anforderung-, Funktions- oder Systemstrukturen. Die Modelle sind beschreibend und können nicht simuliert werden. Als Autorenwerkzeuge dienen z.B. Editoren für Beschreibungssprachen wie SysML.
- Modellbildung und erste Simulation: Auf dieser Ebene werden meist quantitative, simulierbare Modelle erstellt, z.B. multiphysikalische Simulationsmodelle die mehrere Disziplinen mit einbeziehen. Als Autorenwerkzeuge dienen z.B. Simulationseditoren wie Dymola oder Matlab/Simulink.
- Disziplinspezifische Modellbildung: Auf dieser Ebene werden z.B. Geometrie- oder CAE-Modelle erstellt, die einen sehr disziplinspezifischen Charakter haben. Als Autorenwerkzeuge dienen z.B. CAD Systeme oder disziplinspezifische Berechnungs- und Simulationssoftware.

Die Anforderungsdefinition ist der Ausgangspunkt der Entwicklung. Es spiegelt die mehr oder weniger abstrakte Idee in Form von Kundenbedürfnissen oder Anforderungen der Anwender wieder. In der folgenden Anforderungsanalyse werden die Kundenanforderungen in logisch konsistente, technische Anforderungen übersetzt. Dies ist mit A in Abbildung 2 markiert.

In der frühen Phase der Produktentwicklung ist der interdisziplinären Systementwurfs für die Erstellung einer funktionalen Lösung, die alle Disziplinen vertreten können, unerlässlich. Dies wird auch oft als Requirements Engineering bezeichnet. Beginnend mit einer gro-

ben Funktions- und Verhaltensbeschreibung, kann dieses Konzept Schritt für Schritt verfeinert werden. Die Aufgliederung und Abbildung in Funktionen und Teilfunktionen wird mit F in Abbildung 2 markiert. Diese bietet eine zunächst disziplin- und lösungsneutrale Sichtweise auf das Gesamtsystem in Form einer ersten Spezifikation.

Das Lösungskonzept wird durch die Definition von logischen Komponenten (gekennzeichnet mit L in Abbildung 2) die funktionale Elemente und Verhalten realisieren, beschrieben. Das Lösungskonzept umfasst das logische und physikalische Verhalten sowie die Struktur des Systems.

Semi-formale Modellierungssprachen, wie UML oder SysML, sowie simulationsbasierte Modellierungssprachen wie Matlab/Simulink/Simscape oder Modelica unterstützen die interdisziplinäre Systementwicklung, so dass am Ende der interdisziplinären Systementwicklung gezielte Produkteigenschaften durch virtuelle Tests überprüft werden können. Bisher gibt es keinen durchgängigen Daten- bzw. Informationsaustausch zwischen der modellbasierten Spezifikation über die ersten Simulationen zu den entsprechenden Disziplinen. Gerade eine inkrementelle und integrierte Überprüfung von Eigenschaften mit virtuellen Tests setzt die Entwicklung von Anforderungen und definierten Testszenarien voraus.

Basierend auf den ersten Simulationen und der funktionalen Beschreibung beginnt die disziplinspezifische Entwicklung, die die physischen Elemente des Systems, wie Hardware-Teile oder Software-Code (gekennzeichnet mit P in Abbildung 2) adressiert. Hier setzt meist die CAx Prozesse in der die virtuelle Produktentwicklung an.

5 Datenmanagement für die konzeptionelle Produktmodellierung

Eine vollständige Definition der Anforderungen, Funktionen und logischen System-elementen ist schwer schon zu Beginn zu erreichen. Daher sollte der Entwicklungsprozess nach dem oben vorgestellten V-Modell in inkrementellen Schleifen verlaufen, die alle Aspekte weiter verfeinern. Die Iterationen beginnen mit dem kleinsten »V«,

bis es über die virtuellen Test-Iterationen mit detaillierten Simulationsmodellen, zur physikalischen Prüfung kommt. Jede Iteration bedeutet eine Zunahme des Wissens über das Produkt. Gleichzeitig bedeutet dies, dass Änderungen vorgenommen werden können, die entsprechend verwaltet werden müssen. Hierfür bietet sich das PDM System als Kern einer PLM Lösung an, dort wo zentrale Produktinformationen aus dem Entwicklungsprozess von allen beteiligten Personen zusammen getragen und organisiert werden.

Modelle in der Sicht der Modellierung und Spezifikation (Abbildung 2, oben links) werden in den frühen Phasen der Produktentwicklung mit dem Zweck erstellt, die Beschreibung des Produktes zu verwalten und alle kritischen Aspekte unter Einbeziehung aller Beteiligten zu überprüfen.

Basierend auf dem System-Spezifikationsmodell können verschiedene Aspekte des Produktes »eingefroren« und »freigegeben« werden, so dass in der disziplinspezifischen Entwicklung darauf Bezug genommen werden kann. Dies können z.B. Parameter von Komponenten für bestimmte disziplinspezifische Planungen und Simulationen sein, die wiederum auch für andere disziplinspezifische Modelle verwendet werden.

Die Verwaltung von funktionalen Beschreibungsmodellen in PDM kann während der frühen Planungsphasen, sowie später entlang des Produktlebenszyklus als Medium für die Rückverfolgung von Änderungen an Anforderungen, Funktionen und Verhalten, logischen sowie physikalischen Elementen sein. Ein Beispiel hierfür ist die Möglichkeit, durch die funktionale Beschreibung eine Zuordnung zu schaffen, welche Anforderungen eine Änderung in der Produktstruktur beeinträchtigt und umgekehrt.

5.1 Computerunterstützung für die funktionale Produktbeschreibung

Die funktionale Produktbeschreibung bezieht sich auf die Sicht der Modellierung und Spezifikation (Abbildung 2, oben links) und beschreibt das System aus einer funktionsorientierten Perspektive. Dazu gehören die Anforderungen, die funktionalen und logischen Elemente des Systems. SysML eignet sich für die Modellierung die-

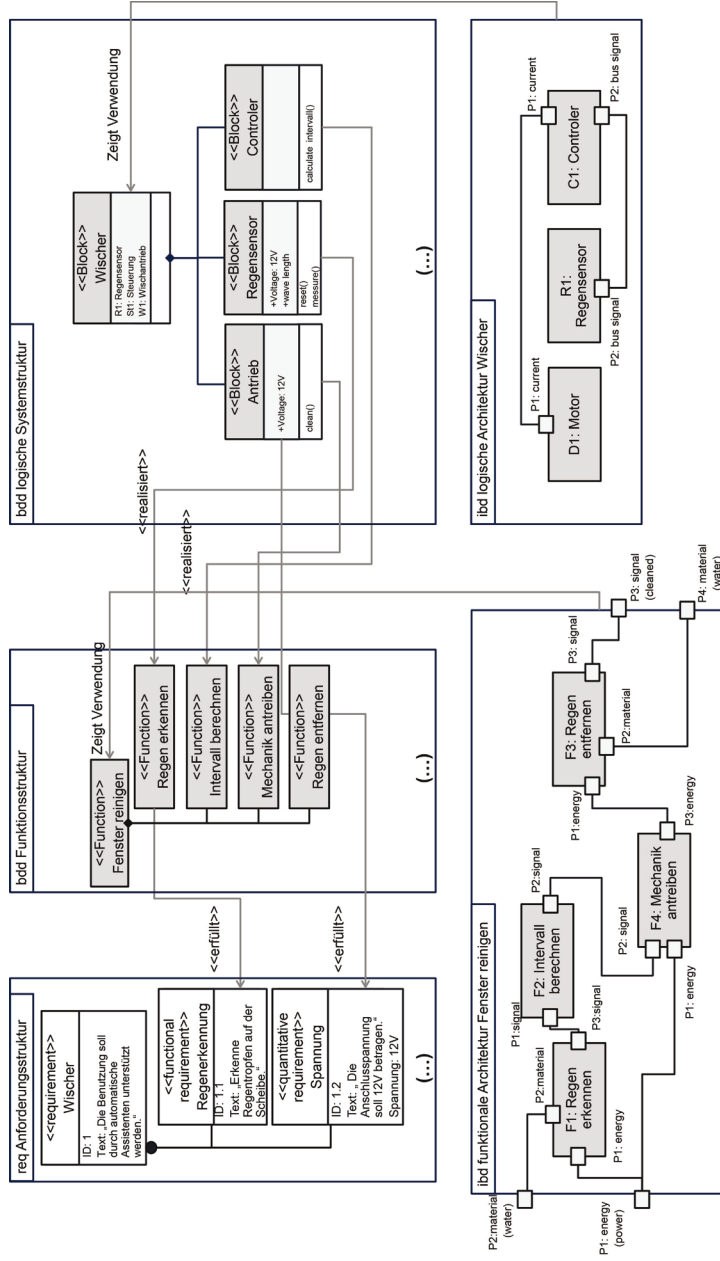


Abbildung 3: Beispiel einer funktionalen Produktbeschreibung

ser Aspekte. SysML ist eine standardisierte Modellierungssprache und es sind viele Werkzeuge für die Modellierung existent. XML stellt hier eine Basis für den Datenaustausch, ist jedoch von vielen Werkzeugen nicht vollständig unterstützt. Zusätzlich zu den SysML-Modellierungs-Tools wie z.B. Magicdraw oder Enterprise Architect, bieten Anforderungsmanagement-Tools wie z.B. Doors, Requisite-Pro oder MKS starke Unterstützung für die Modellierung von Anforderungen an.

Abbildung 3 zeigt ein exemplarisches Beispielmodell eines Scheibenwischers in SysML. Die Anforderungen sind in einem hierarchischen Anforderungsdiagramm modelliert. Funktionen und logische Elemente des Systems sind hierarchisch in einem Blockdefinitionsdiagramm modelliert. Die interne Struktur wird durch interne Blockdiagramme abgebildet, die die funktionale und logische Architektur repräsentieren.

Zwischen zwei verschiedenen Modellelementen, kann eine Zuordnungsbeziehung aufgebaut werden. Die so genannten Querverweise können auf verschiedene Arten eingeschränkt werden. In SysML existieren vordefinierte Zuweisungen, darüber hinaus können aber auch neue Zuweisungen definiert werden. In dem angeführten Beispiel ist die Zuordnung »realisiert« als eine Beziehung zwischen logischen und funktionalen Elementen des Systems verwendet. Der Regensensor realisiert die Funktion »erkenne Regen«. Ein Merkmal (Property) eines Systemelements »erfüllt« eine Anforderung. In unserem Vorschlag wird diese Zuordnung erweitert, so dass auch eine Funktion eine Anforderung erfüllen kann. Sind diese Querverweise definiert, können Anforderungen mit der funktionalen und logischen Architektur in Zusammenhang gebracht werden.

Soll der Scheibenwischer in Abbildung 3 in einem Lkw verwendet werden, kann es sein, dass die Bordspannung 24 V beträgt und sich somit die Anforderungen verändert. Um diese veränderte Anforderung umzusetzen ist es wichtig alle betroffenen logischen oder funktionalen Elemente des Systems zu kennen.

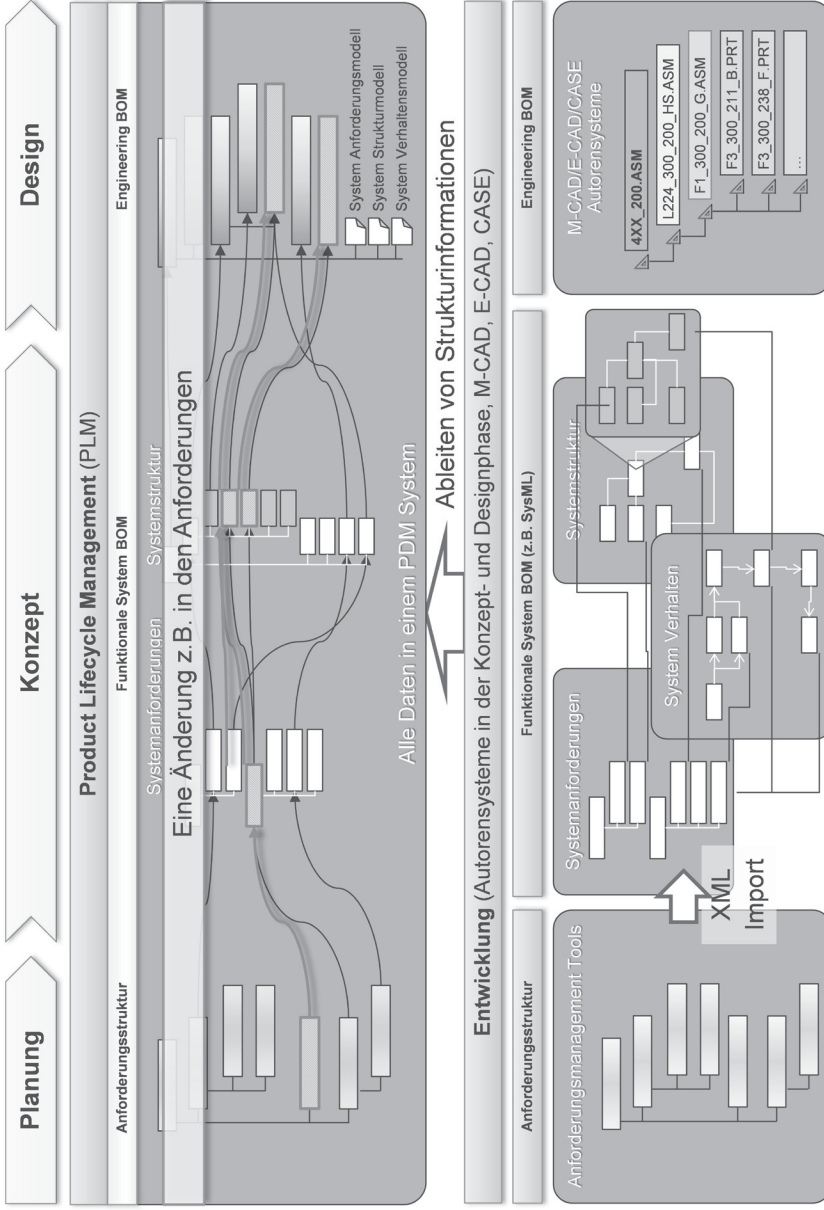


Abbildung 4: funktionale Produktbeschreibung als Teil eines PLM Konzepts

5.2 Inhalt eines funktionalen Produktbeschreibungsmodells

SysML-Editoren können helfen, ein erstes Konzept eines Systems mit Hilfe von unterschiedlichen Diagrammen grafisch zu modellieren. Zur Integration der modellierten Daten in ein PDM System wird im Folgenden ein vereinfachtes Datenschema zur Abbildung auf z.B. XML vorgeschlagen, das unterschiedliche Perspektiven beinhaltet:

- *Hierarchien*: Anforderungs-, Funktions- und logische Elementstrukturen können in einer hierarchischen Weise betrachtet werden. In PDM-Systemen können diese in einfacher Form als Struktur (ähnlich einer Stückliste) gespeichert werden.
- *Querverweise unter Modellelementen und über Modellgrenzen hinweg*: Querverweise sind die Beziehungen zwischen verschiedenen Arten von Modell-Elementen, die mit Zuweisungen in SysML modelliert werden können. Der Vorteil des modellbasierten Systems Engineering ist, dass diese Querverweise zusammen mit dem Systemelement verwaltet werden.
- *Typisierte interne Verbindungen unter Modellelementen innerhalb einzelner Modelle*: Schematische Blockdiagramme, z.B. in Modelica oder interne Blockdiagramme in SysML, beschreiben die interne Struktur eines Systemelements über Verbindungen zwischen zwei Ports gleichen Typs. Dies stellt die interne Vernetzung der transparent dar.

Hierarchien werden in erster Linie für das Management komplexer Systeme verwendet. Werden funktionale oder logische Zusammenhänge betrachtet ist dies für eine funktionale Produktbeschreibung meist nicht ausreichend.

5.3 Datenmodell für das Produktdatenmanagement

Abbildung 4 zeigt die mögliche Integration einer funktionalen Produktbeschreibung in eine PLM Umgebung. Für diesen Beitrag wird davon ausgegangen, dass die Anforderungen und die Stücklistenstruktur im PDM-System bereits vorhanden sind. Ein Autorenwerkzeug hierfür ist z.B. Doors. Die funktionale Produktbeschreibung integriert

sich zwischen Anforderungen und der Stücklistenstruktur (BOM). Funktionen lassen sich auf einfache Art und Weise hierarchisch modellieren (siehe Abschnitt 3.1). Dies kann in SysML-Editoren über Blockdefinitionen modelliert und in XML extrahiert werden. Logische Systemelemente stellen die Realisierung von Funktionen mit einem physikalischen Effekt dar und definieren Eigenschaften. Ein Systemelement kann weitere externe Modelle als Dateien referenzieren. Dies könnte zum Beispiel eine Modelica oder Matlab/Simulink/Simscape Modelldatei sein, die entwickelt wurde, um die physikalische Eigenschaften darzustellen und zu analysieren. Ein logisches Systemelement ist mit der physikalischen Stückliste (BOM) verbunden, welche M-CAD oder E-CAD Dateien, sowie Software aufnimmt.

6 Zusammenfassung und Ausblick

In diesem Beitrag wurde eine Erweiterung des V-Modells nach VDI 2206 vorgeschlagen, welches auf die Herausforderungen der virtuellen modellbasierten Produktentwicklung näher eingeht. Des Weiteren wurde ein Datenmodell für die funktionale Produktbeschreibung vorgestellt, welches einen leichteren Zugang zu den Methoden des modellbasierten Systems Engineering für Organisationen ermöglichen soll, um eine interdisziplinären Produktentwicklung zu unterstützen. Der Anwendungsbereich ist auf hierarchische und interne Strukturen, sowie Querverweise zwischen Modell-Elementen fokussiert. Eine Integration der funktionalen Produktbeschreibung in ein PLM Konzept beinhaltet die Verfolgung von Änderungen und Einflüssen auf Anforderungen, Funktionen und logisches Systemelementen. Ein Management der funktionalen Systembeschreibung ermöglicht erst die frühe Produktdokumentation und Qualitätssicherung. Das vorgeschlagene Vorgehen in dem erweiterten V-Modell ist angelehnt an das modellbasierte Systems Engineering, was ein grundsätzliches Umdenken in der Produktentwicklung erfordert. Das vorgeschlagene Datenmodell implementiert drei Perspektiven: Hierarchien, Querverweise zwischen Modellelementen und typisierte Verbindungen innerhalb der Modellelemente. Diese Perspektiven sollten das Management der funktionalen und logischen Architekturen in einer PLM-Lösung zu ermöglichen.

Literaturverzeichnis

- Anderl, R, Eigner, M, Sendler, U, Stark, R (2012): Smart Engineering – Interdisziplinäre Produktentstehung, in: acatech diskussion, Heidelberg: Springer
- Broy, M, Glotzbach, U (2010): Cyber-Physical Systems - Innovation durch software-intensive eingebettete Systeme. In: acatech diskutiert, Heidelberg: Springer
- Eckert, R, Mansel, W, Specht, G (2005): STEP AP233 + Standard PDM = Systems Engineering PDM?, in: Proceedings of 11th international Conference on Concurrent Engineering, München, 20.-22.06.2005, 405–412.
- Friedenthal, S, Steiner, R, Moore, A (2009): A Practical Guide to SysML - The Systems Modeling Language, San Francisco: Morgan Kaufmann Pub
- Gausemeier, J, Frank, U, Donoth, J, Kahl, S (2009): Specification technique for the description of self-optimizing mechatronic systems, in: Research in Engineering Design, 20 (4), Heidelberg: Springer, 201–223.
- Hirtz, J, Stone, R, McAdams, D, Szykman, S, Wood, K (2002): A functional basis for engineering design: Reconciling and evolving previous efforts, In: Research in Engineering Design, (13), Heidelberg: Springer, 65–82
- Pahl, G, Beitz, W, Feldhusen, J, Grote, K-H (2007): Konstruktionslehre - Grundlagen erfolgreicher Produktentwicklung . 7. Aufl., Heidelberg: Springer
- Paredis, C, Bernard, Y, de Koning, H-P, Friedenthal, S, Fritzon, P, Rouquette, N, Schamai, W (2010): An Overview of the SysML-Modelica Transformation Specification, Beitrag im Internet: http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-modelica:sysml_and_modelica_integration veröffentlicht 2010, abgerufen am 02.02.2012.
- Schamai, W, Fritzon, P, Paredis, C, Pop, A (2009): Towards Unified System Modeling and Simulation with ModelicaML: Modeling of Executable Behavior Using Graphical Notations, in: Proceedings 7th Modelica Conference, Como/Italy, 20.–22.09.2009, 612-621.

Kontakt

Prof. Dr.-Ing. Martin Eigner
 Dipl.-Ing. Torsten Gilz
 Dipl.-Ing Radoslav Zafirov
 Technische Universität Kaiserslautern
 Lehrstuhl für Virtuelle Produktentwicklung
 Gottlieb-Daimler-Straße 44
 67663 Kaiserslautern
<http://vpe.mv.uni-kl.de>

